2004

A GCD attack resistant CRTHACS for secure group communications

Xukai Zou
Indiana University-Purdue University Indianapolis, xkzou@cs.iupui.edu

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Spyros S. Magliveras
Florida Atlantic University

Proceedings. ITCC 2004. International Conference on Information Technology: Coding and Computing, 2004.
Volume: 2
Digital Object Identifier: 10.1109/ITCC.2004.1286618
Publication Year: 2004 , Page(s): 153 - 154 Vol.2

A GCD attack resistant CRTHACS for secure group communications

Xukai Zou
School of Science, Purdue University -Indianapolis
Indianapolis, IN 46202, USA
xkzou@cs.iupui.edu

Byrav Ramamurthy
University of Nebraska-Lincoln
Lincoln, NE 68588, USA
byrav@csce.unl.edu

Spyros S. Magliveras
Florida Atlantic University
Boca Raton, Florida 33431, USA
spyros@fau.edu

Abstract

In this paper, we propose an improved CRTHACS scheme for secure group communications. The scheme resists several GCD attacks which exist in the original CRTHACS scheme [2] and were recently reported in [1].

1. Introduction

Secure group communication (SGC) with hierarchical access control (HAC) refers to a scenario where a group of members is divided into different privileged subgroups located at different levels and a high-level subgroup can receive and decrypt messages within any of its descendant lower-level subgroups, but the converse is not allowed.

In [2], a Chinese Remainder Theorem based HAC scheme (CRTHACS) for SGC was proposed. The scheme was intended not only to enforce HAC but also to operate without disclosing the hierarchy and the receivers. However, some recent attacks, presented in [1], and based on computing certain *greatest common divisors* (GCDs), have been shown to disclose the hidden hierarchy. In this paper, we propose a solution to defeat these GCD attacks, thus keeping the hierarchy hidden.

In sections 2 and 3, we briefly summarize the CRTHACS scheme from [2] and three kinds of GCD based attacks from [1], respectively. We describe our solution for countering the attacks in section 4.

Throughout this paper, $E_k(x)$ denotes public key encryption under key k and $\{x\}_k$ secret key encryption under key k.

2. CRTHACS for SGC

In CRTHACS there is a Group Controller (GC) and subgroups G_i, as well as a subgroup controller for each G_i, which will also be denoted by G_i. The GC has a pair of public and private keys (P_{GC}, S_{GC}), and so does each subgroup G_i, denoted by (P_i, S_i). The GC maintains the entire hierarchical structure of the group; generates a random set of pairwise relatively prime numbers $N_0, N_1, N_2, \cdots, N_r$, where N_0 is public, while the remaining N_i are secret. There are also positive integers $\mathcal{N}_i$, one for each subgroup G_i, defined by equation (2). The GC computes $\mathcal{N}_i$ and COM_CRT_i and sends $\mathcal{N}_i$, COM_CRT_i and N_i to G_i by means of a secure channel. Let $\{G_{i_1}, G_{i_2}, \cdots, G_{i_k}\}$ be the collection of all ancestral subgroups of G_i, and consider below the system of congruences (1) and equation (2) which define COM_CRT_i and $\mathcal{N}_i$ respectively.

$$\begin{array}{rcl} COM\_CRT_i & \equiv & E_{P_{i_1}}(K_i) \; mod \; N_{i_1} \\ COM\_CRT_i & \equiv & E_{P_{i_2}}(K_i) \; mod \; N_{i_2} \\ & \vdots & \\ COM\_CRT_i & \equiv & E_{P_{i_k}}(K_i) \; mod \; N_{i_k} \end{array} \tag{1}$$

$$\mathcal{N}_i = N_{i_1} \cdot N_{i_2} \cdots N_{i_k} \tag{2}$$

To every subgroup G_i a six-tuple $(P_i, S_i, K_i, N_i, COM\_CRT_i, \mathcal{N}_i)$ is assigned, where K_i is the secret data communication key for G_i. K_i is sent securely to the GC by G_i and except for P_i, the remaining five elements are kept secret, known only by G_i. Besides knowing its subgroup's six elements, every participant j has its own public and private key (p_j, s_j).

Whenever a participant j with identity ID_j in G_i sends a message M, it broadcasts the tuple $(ID_j, CRT_i, \{M\}_{K_i})$ where CRT_i is computed as:

$$\begin{array}{rcl} CRT_i & = & COM\_CRT_i \; mod \; \mathcal{N}_i \\ CRT_i & = & E_{s_j}(MAC_{K_i}(\{M\}_{K_i})) \; mod \; N_0 \end{array} \tag{3}$$

When a receiver m receives $(ID_j$, CRT_i, $\{M\}_{K_i})$, it can check whether the message is intended for itself, verify both the sender and the message, and decrypt the message if the check and verification succeed or discard the message otherwise.

3. GCD based attacks

In [1], the authors put forth three possible attacks, all based on computing a number of GCD's, which make possible the disclosure of the hierarchy.

1. Note that CRT_i is dependent on messages, but COM_CRT_i is *not*. In particular, from the first equation of (3) any two instances of CRT_i will differ by a multiple of $\mathcal{N}_i$. Therefore from the CRT_i's of more than two messages of the same subgroup G_i, an attacker (outside G_i) can derive information about $\mathcal{N}_i$, as $CRT_{i_j} - CRT_{i_k} \equiv 0 \bmod \mathcal{N}_i$. Hence, $gcd(CRT_{i_1} - CRT_{i_2}, CRT_{i_1} - CRT_{i_3})$ divides $\mathcal{N}_i$. If the attacker has several CRT_{i_j} at his disposal he refines this information about $\mathcal{N}_i$.
2. The subgroup G_i may figure out its ancestor G_j (i.e., N_j) by computing $gcd(\mathcal{N}_i, COM\_CRT_i - E_{P_j}(K_i))$ since G_i possesses $\mathcal{N}_i$ and COM_CRT_i and can compute $E_{P_j}(K_i)$.
3. If two subgroup G_i and G_j collude, they may find their common ancestor G_k (i.e., N_k) by computing $gcd(CRT_i - E_{P_k}(K_i), CRT_j - E_{P_k}(K_j))$.

4. Improved CRTHACS

The solution to the three attacks can be summarized as three modifications to the original CRTHACS scheme:

1. Move the signed MAC out of CRT_i and send the signed MAC as a separate item. Thus, CRT_i will be independent of messages. This will beat the attack of the first kind.
2. Remove COM_CRT_i and replace CRT_i with a new CRT_i. Moreover the new CRT_i will now be computed by the GC. Instead of sending COM_CRT_i and $\mathcal{N}_i$ to G_i, the GC just sends CRT_i to G_i. G_i uses CRT_i directly but does not know $\mathcal{N}_i$. Thus, the second attack is defeated.
3. Replace the public encryption of $E_{P_{i_k}}(K_i)$, corresponding to its ancestral subgroup G_{i_k}, with a secret key encryption $\{K_i\}_{K_{i_k}}$ in the congruence system (1). As a result, G_i cannot compute $\{K_i\}_{K_k}$ related to its ancestral subgroup G_k because G_i does not know K_k. This defeats the third type of attack. In fact, this modification also defeats the second attack since $E_{P_j}(K_i)$ is not involved in COM_CRT_i.

We describe the modifications in detail. The COM_CRT_i and $\mathcal{N}_i$ are removed from the scheme and CRT_i is computed by the GC and sent to G_i directly. The system of equations (1) is replaced by the system (4) as follows:

$$\begin{array}{lll} CRT_i & \equiv & \{K_i\}_{K_{i_1}} \ mod\ N_{i_1} \\ CRT_i & \equiv & \{K_i\}_{K_{i_2}} \ mod\ N_{i_2} \\ & \vdots & \\ CRT_i & \equiv & \{K_i\}_{K_{i_k}} \ mod\ N_{i_k} \\ CRT_i & \equiv & \{K_i\}_{K_i} \ mod\ N_0 \end{array} \quad (4)$$

The $N_0, N_1, \ldots, N_r$ are as in the original CRTHACS. Every subgroup controller G_i will no longer have six but five elements: $(P_i, S_i, K_i, N_i, CRT_i)$ and every participant $k \in G_i$ will also have five elements $(p_k, s_k, K_i, N_i, CRT_i)$. Initially, the GC sends N_i and CRT_i to the subgroup controller G_i securely, and then G_i multicasts the two values to all participants in its subgroup.

Whenever participant $k \in G_i$, with identity ID_k, sends a message M, it computes and sends (ID_k, CRT_i, $Signed\_MAC$, $\{M\}_{K_i}$), where $Signed\_MAC = E_{s_k}(MAC_{K_i}(\{M\}_{K_i}))$. As indicated above, the $Signed\_MAC$ is sent as a separate item.

Assume that sender $k \in G_i$. When a receiver m receives (ID_k, CRT_i, $Signed\_MAC$, $\{M\}_{K_i}$), it proceeds as follows: (1) If $m \in G_i$, then m has the same key K_i as sender k. If $m \in G_j$ where G_j is an ancestral subgroup of G_i, m computes $t = CRT_i \ mod\ N_j$ (i.e, $\{K_i\}_{K_j}$) and decrypts t to get K_i, (2) m computes $x = \{K_i\}_{K_i}$ and $y = CRT_i \ mod\ N_0$, (3) m compares x and y; if $x \neq y$, the verification of the key fails (there are two possibilities: the CRT_i was modified during transmission or the receiver is not in the sender's subgroup or its ancestral subgroups). The receiver discards the message. Otherwise (i.e., $x = y$), the key is correct and the message is intended for m, (4) Decrypts the $Signed\_MAC$ using k's public key to get $MAC_{K_i}(\{M\}_{K_i}) = E^{-1}_{p_k}(Signed\_MAC)$, where E^{-1} stands for the decryption algorithm corresponding to E, (5) Computes $MAC_{K_i}(\{M\}_{K_i})$ using K_i (which already passed the verification in (3)), (6) Compares the above two MACs. If the two MACs are equal, then both the sender and the message are authenticated. The receiver decrypts the message using K_i. Otherwise, the message was modified during transmission, and the receiver discards it.

It is worth pointing out that the improved CRTHACS has an extra advantage over the original one, viz., better efficiency because computing CRT_i and K_i will not involve public key encryption/decryption operations.

5. Conclusion

In this paper, we have proposed a solution which defeats a number of recent attacks, satisfies all original goals, and provides better performance.

References

[1] R. Steinwandt and W. Geiselmann. Attacks on a secure group communication scheme with hierarchical access control. *Submitted to International Conference on Information Security and Cryptography, Seoul, Korea*, November 2003.

[2] X. Zou, B. Ramamurthy, and S. Magliveras. Chinese remainder theorem based hierarchical access control for secure group communications. *Lecture Notes in Computer Sciences (LNCS), Springer-Verlag, (International Conference on Information and Communication Security)*, 2229:381–385, 2001.